FOR ARGUMENT

Supreme Court, U. S.
FILED
APR 7 1978
MICHAEL RODAK, JR., CLERK

IN THE

Supreme Court of the United States

October Term, 1977.

No. 77-642.

LUTRELLE F. PARKER, Acting Commissioner of Patents and Trademarks,

Petitioner,

v.

DALE R. FLOOK,

Respondent.

On Writ of Certiorari to the United States Court of Customs and Patent Appeals.

BRIEF AMICUS CURIAE FOR THE ASSOCIATION OF DATA PROCESSING SERVICE ORGANIZATIONS (ADAPSO).

MORTON C. JACOBS,
DAVID COHEN,
1420 Walnut Street,
Philadelphia, Pa. 19102
Attorneys for Amicus ADAPSO.

International Printing Co., 711 So. 50th St., Phila., Pa. 19143 — Tel. (215) 727-8711

TABLE OF CONTENTS.

	Page
QUESTIONS PRESENTED	2
INTEREST OF AMICUS CURIAE	2
SUMMARY OF ARGUMENT	4
ARGUMENT	7
I. The Writ of Certiorari Should Be Dismissed	7
A. The Issue Before the Court Has Been Erroneously Presented as Involving the Question of Patentable Subject Matter Under Section 101	7
1. The Factual Premise Upon Which the Grant of Certiorari Is Based Is Mooted	7
2. A Section 103 Inquiry Is Implicitly Raised for the First Time Before This Court by the Patent Office	7
B. Also Erroneously Presented as Before the Court Is the Question of Patents for Computer Programs or "Computer Applications"	13
1. The Record in This Case Is Insufficient to Permit Consideration of the Questions Presented by the Government	13
2. Improperly Presented to the Court Is the Request for a Rule Declaring Machines or Processes Built With Computer Programs to Be Unpatentable Subject Matter	15
II. Special-Purpose Computers Built by Programming a General-Purpose Computer Form Machines and Machine Processes, Hence Patentable Subject Matter	16
A. The Technology of Computers and Computer Programs	16
1. Respondent's Programmed Computer for Automating an Oil Refinery Is a Circuit Configuration	19

TABLE OF CONTENTS (Continued).

	Page
2. The General-Purpose Computer Is an Incomplete Machine That Serves as "a Storehouse of Parts" and Is Completed by Respondent's Stored Program to Form a New Combination of Circuits as Respondent's Programmed Computer for Automating an Oil Refinery	24
3. Hardware and Software Programming Are Equivalent and Alternative Approaches for the Construction of Computerized Control Systems and Equally Patentable	27
4. Software Is Not an "Abstract Idea" and Its Construction Is Engineering	32
III. Rule of Benson	35
IV. Nature of the Software Products Industry	37
A. The Delayed Development of Competition in Software	37
B. The Resulting Poor-Quality Software	40
C. The Small Company Potential for Competition	43
D. The Importance of Patent Incentives	44
V. Equal Protection Requires That Manufacturers of Software Products Be Given the Same Patent Protection Afforded the Hardware Manufacturers	46
CONCLUSION	52

TABLE OF CITATIONS.

Cases:	Page
Alton v. Alton, 347 U. S. 610	8
Bolling v. Sharpe, 347 U. S. 497 (1954)	46
Boyden v. Comm'r. of Patents, 441 F. 2d 1041 (D. C. Cir. 1971)	47
Burr v. Duryee, 68 U. S. 531	46
Dann v. Chatfield, No. 76-1559, cert. den. — U. S. —	15, 27
Dann v. Johnston, 425 U. S. 219	15, 17, 18, 27, 28, 35, 36, 48
Dann v. Noll, No. 76-1558, cert. den. — U. S. —	15, 27
Digitronics Corp. v. The New York Racing Association, 187 USPQ 602, 553 F. 2d 740	31
Glidden Co. v. Zdanok, 370 U. S. 530	12, 16
Gottschalk v. Benson, 409 U. S. 63	15, 27, 32, 36
Graham v. John Deere Co., 383 U. S. 1	9, 10, 11
In re Bernhart, 417 F. 2d 1393 (1960)	14, 17
In re Comstock, 418 F. 2d 905 (1973)	17
In re Knowlton, 481 F. 2d 1357 (1973)	17
In re Prater & Wei, 415 F. 2d 1393 (1969)	14, 17, 19, 24, 25
Johnson v. Massachusetts, 390 U. S. 511	13
Kimball v. Kimball, 174 U. S. 158	8
Mackay Radio & Tel. Co. v. Radio Corp., 306 U. S. 86	8, 10
Massachusetts v. Painten, 380 U. S. 560	13
Richardson v. Belcher, 404 U. S. 78 (1971)	46
Taylor v. McElroy 360 U. S. 610	8
U. S. v. IBM, 69 Civ. 200 (S. D. N. Y.)	27, 37, 38

Statutes:

15 U. S. C. § 631	47
28 U. S. C. § 1256	12
35 U. S. C. § 100(b)	36
35 U. S. C. § 101	8, 9, 11, 16, 31, 46
35 U. S. C. § 102	27
35 U. S. C. § 103	5, 10, 11, 12, 13, 16, 27
35 U. S. C. § 112	13, 17
35 U. S. C. § 141 et seq.	4, 12

TABLE OF CITATIONS (Continued).

Statutes (Continued): Page

35 U. S. C. § 144 ..... 12, 13
42 U. S. C. § 281(a) 16
42 U. S. C. § 2181 46
42 U. S. C. § 2187(b), (3) 46

Miscellaneous:

Berteau, "IBM and the Structure of the Industry, The Software Monopoly", 21 Datamation 111 (October, 1975) ... 38
Brenner, Commissioner of Patents, "The Future of Computer Programs in the U. S. Patent Office; Law of Software (1968) 18, 45
Bronowski, J. "The Ascent of Man" (1973) 18
Brown, "Smoothing, Forecasting & Prediction of Discrete Time Series" (1963) 7
Burks, Goldstine, von Neumann "Preliminary Discussion of the Logical Design of an Electronic Computing Instrument" (1946) 29
"Composite Design: The Design of Modular Programs," IBM Report (Jan. 1973) 33
Computer Software Management. A primer for project management and quality control—Dennis W. Fife. U. S. Dept. of Commerce; NBS Special Publication 500-11, July 1977 44
Computerworld, November 20, 1974 40
Computerworld, August 6, 1975 40
"Customers' Search for Software Substitutes" Keystone Address at ADAPSO's 46th Management Conference "Responding to Change" April 18, 1977 42
"EDP—A 20 Year Ripoff," INFOSYSTEMS, 11/74 42
Eves, H., "An Introduction to the History of Mathematics" (1976) 8
Frost & Sullivan Computer Software & Packaged Services Market (Dec. 1977) 39

TABLE OF CITATIONS (Continued).

Miscellaneous (Continued): Page

Goldstine, "The Computer from Pascal to von Neumann" (1972) 26, 37, 47
Guidelines for Examination of Computer Programs (1966); 829 O. G. 442; 855 O. G. 829 ..... 17, 25, 26, 45
H. R. 5924, proposed, 35 U. S. C. § 106, 90th Cong., 1st Sess. (1967) 50
"Hierarchies—Software Engineering Design," Software Engineering, International Computer State of the Art Report (1972) 32
Iberall, A., "Toward a General Science of Viable Systems" (1972) 19
IBM report, "Composite Design: The Design of Modular Programs" (Jan., 1973) 42
"Management Concepts for Top-Down Structural Programming," IBM Report (Feb. 1973) 33
Manual of Patent Examining Procedure, Section 1216.01(4) . 12
National Bureau of Standards Experimental Technology Inventives Program, Preliminary Program Description, Fiscal Year 1973 Budget Submission 43
New Encyclopedia Britannica 22, 29, 31
"155, 165 Owners Angry With IBM," Datamation, August, 1974 42
1966 Report of the President's Commission on the Patent System (reprinted in S. Doc. No. 5, 90th Cong., 1st Sess.) 48
Prof. Ivan Flores, City Univ. N. Y., "Computer Software", Science & Technology No. 89, May 1969 ..... 17, 25
Professor Ivan Flores, "Computer Logic; the Functional Design of Digital Computers" (1960) Prentice-Hall, Inc. . . 19, 22
Prof. Saul Gorn, Final Report No. AD 59 URI, Univ. of Pa. Moore School of Electrical Engineering to U. S. Army Signal Research & Development Laboratories (1959) . . 15, 28
Professor Saul Rosen, "Hardware Design Reflecting Software Requirements," 33 Proc. Fall Joint Computer Conf. (1968) 22, 28

TABLE OF CITATIONS (Continued).

Miscellaneous (Continued): Page

Report, Brooks Subcommittee on H. R. 4845, Aug. 17, 1965, 89th Cong. 1st Sess. (House Rep. No. 802) "Domination of ADP Industry by IBM" 37

I Robinson on Patents § 173 24

Rosen, S., "Electronic Computers: A Historical Survey", Computer Surveys (1969) 29

Science, American Association for the Advancement of Science, Vol. 195, No. 4283, 18 March 1977 41

Scientific American (May 1975) 23

"'Software Dinosaur' Slows Growth," Univac Vice President Neil Gorchow, Sperry Rand, Software Digest, Vol. 7, No. 20, May 22, 1975 42

"Software Engineering," Report on a World Conference Sponsored by the NATO Science Committee, Garmisch, Germany (Jan., 1969) 35

Subtests of "Nonobviousness" 112 U. Pa. L. Rev. 1969 (1964) 11

"Top-Down, Bottom-Up, Structured Programming & Program Structure," Draper Labs, MIT (Dec., 1972) 34

Turn, R., "Computers in the 1980's," Columbia Univ. Press (1974) 34

U. S. Patent No. 3,873,982, "Analytic Character Recognition System" 27

Vacroux—"Microcomputer" 29, 30

Wilkes, M. V., "Progress in High Speed Calculating Machine Design" (1949) 30

IN THE

Supreme Court of the United States

OCTOBER TERM, 1977

No. 77-642

LUTRELLE F. PARKER,
ACTING COMMISSIONER OF PATENTS AND TRADEMARKS,
Petitioner,

v.

DALE R. FLOOK,
Respondent.

ON WRIT OF CERTIORARI TO THE UNITED STATES COURT OF CUSTOMS AND PATENT APPEALS

BRIEF AMICUS CURIAE FOR THE ASSOCIATION OF DATA PROCESSING SERVICE ORGANIZATIONS (ADAPSO)

The Association of Data Processing Service Organizations files this brief amicus curiae in opposition to Petitioner, and also files letters of consent of the parties in accordance with the Rules.

QUESTIONS PRESENTED.

1. Whether the writ of certiorari should be dismissed in a case in which (a) the factual premise for requesting certiorari is mooted, (b) there is a jurisdictional barrier to the consideration of a central issue, (c) the factual record is inadequate and (d) a principal question for which certiorari was requested is improperly presented to the Court?

2. Whether an otherwise patentable automated process becomes unpatentable subject matter solely because a portion of the process is performed by a programmed general-purpose computer?

3. Whether an otherwise patentable process becomes unpatentable subject matter solely because a portion employs a programmed digital computer to perform calculations?

4. Whether software manufacturers have protection of the patent laws equal to that given to hardware manufacturers.

INTEREST OF AMICUS CURIAE.

Association of Data Processing Service Organizations (hereinafter ADAPSO) is a nonprofit trade association of 308 companies in the computer field,[1] approximately 33% of whom concentrate their attention in the software products and services industry and the remainder largely in the furnishing of varied time sharing and data center services.

ADAPSO presents its views to the Court because an inaccurate description of software technology and the software products industry has been set forth by petitioner's brief and the supporting brief of the hardware manufacturers' association, CBEMA.

ADAPSO software product companies are not developers of mathematical algorithms. They are companies engaged in building and manufacturing software machines for which profit-conscious companies will and do expend money. ADAPSO members believe they are entitled to the same patent protection for their innovative products as that granted the hardware manufacturers.

1. The Association has 308 member companies whose annual revenues are broken up as follows:

Revenues	*Number of Members*
$100,000 - $ 1 million	201
$ 1 million - $ 5 million	75
$ 5 million - $10 million	13
$10 million - $50 million	12
$50 million and over	7

Virtually all its member companies develop computer programs and require different forms of protection for these programs, including

1. (Cont'd.)

the availability of patent protection on the same basis currently available to the manufacturers of computer hardware. Generally, member companies belong to at least one of the following three sections of ADAPSO:

a. Software Products and Services Companies Section. Members of this section fall into either one group that offers consulting and customer program services; or another that offers copies of a master program for lease or license on a restricted basis. These copies, along with documentation and support, are referred to as software packages, software products, or program packages.

b. Data Center Section. Members of this section are companies that provide data center computer services to organizations that do not have in-house computers—or that desire specialized outside data processing services.

c. Time-Sharing Section. Members of this section offer remote computer facilities on a national and international basis—usually by providing a terminal and communications lines to their customers.

SUMMARY OF ARGUMENT.

The writ of certiorari should be dismissed as improvidently granted, because the Patent Office objections to the issuance of a patent to Flook are mooted, and because of the inadequate record, the jurisdictional barrier to the consideration of a central factual issue, and the inappropriateness of the principal question for which certiorari was granted. To consider what the Patent Commissioner presents as the key issues in this case would require this Court to undertake a thorough inquiry into the complex technological issues of special-purpose computer construction, but the record does not provide the factual basis which would enable the Court to do so. The Supreme Court is not the appropriate forum for initial fact finding.

The most basic technological issue—the Board of Appeals' finding in a predecessor case that programming a general-purpose computer restructures the computer into a different machine—is not jurisdictionally subject to review because 35 U. S. C. § 141, *et seq.* bind the Commissioner of Patents by Board determinations. Since the Board is the *alter ego* of the Commissioner, his indirect attempt to appeal its finding before this Court is in essence a nonjusticiable intra-agency dispute. Moreover, the C. C. P. A. and the Board have considered the basic technological fact issue, each has special credentials of technological expertise and both have found in favor of the restructuring principle.

The broad question for which *certiorari* was granted —whether innovative machines which can be embodied in computer programs are patentable subject matter—is inappropriate for judicial resolution in this case because patentability only depends on the structural nature of the specific invention, not which of alternative types of construction was employed in a preferred embodiment. The petitioner has conceded that the Court cannot formulate a

rule of general applicability in its decision on this case and that the only other question presented—obviousness under 35 U. S. C. § 103—is jurisdictionally not before the Court.

The petitioner's main legal arguments depend on a mischaracterization of computer program technology. An accurate picture of this technology reveals that a programmed general purpose computer is a different machine than the unprogrammed machine. The designer of software, like any engineer, builds machine structures, not "mathematical algorithms"; he develops machine solutions for machine problems. Computer innovations embodied with stored computer programs are legally entitled to the same patent protection as innovations embodied in wire-connected electronic circuits, for technologically "hardware" and "software" are equivalent.

The *facts* that a programmed computer (what respondent has indicated is one form of his invention) is a machine and the innovative aspect of it is a machine structure cannot be overcome by petitioner's *legal* arguments. A machine or technological process is not a "mathematical algorithm." An innovative machine structure is not the mere embodiment of an abstract idea. Machines and their processes are patentable subject matter and respondent's only argument against software being patentable subject matter is the unsupported statement—appealing perhaps to a layman but universally rejected by the experts—that a program is not a machine but merely an "abstract idea," or a "mathematical algorithm."

Patent protection for software is vital to the health of the data processing industry. In a field dominated by the hardware manufacturers, the growth of the small independent software company has been severely hampered by the bundling practices of the hardware manufacturers which continue to this day in the systems programs area. Only through the availability of protection for their prod-

ucts can the software producers hope to bring competition to an industry that sorely needs it.

The absence of patent protection has forced reliance on trade secrecy and has had the negative effect of deterring disclosure of innovations. The availability of patents, with their public disclosure requirement, would "promote the progress of the useful Arts" in this most sophisticated, pervasive and vital area of technology.

Copyright protection is inadequate and inappropriate to protect innovative machine structures. · Copyright of a particular set of computer language statements does not protect the novel machine structure that is the core of any software invention. The Constitution establishes innovation as the prerequisite for a grant of exclusive rights to machine structures; attempts to protect them via copyright or registration run counter to this principle.

At stake in this case is equal patent treatment for the products of small software companies. The myth perpetuated by the Government and CBEMA—that computer-program embodied inventions are not machine inventions—has been exploded; there is now no justification for the denial of patent protection to software-embodied machines.

ARGUMENT.

I. THE WRIT OF CERTIORARI SHOULD BE DISMISSED.

A. The Issue Before the Court Has Been Erroneously Presented as Involving the Question of Patentable Subject Matter Under Section 101.

1. The Factual Premise Upon Which the Grant of Certiorari Is Based Is Mooted.

The factual premise for certiorari is that the Flook patent application discloses a "novel algorithm" (Pet. 7), that "the sole inventive contribution consists of a mathematical algorithm" (Pet. 2, Question Presented; Pet. 11) and that "[t]he only novelty . . . is in the mathematical formula" (Pet. 19). However, Flook's mathematical formula viewed as an abstract mathematical equation is *not novel;* the same equation appears in the statistical treatise [2] relied on by the Patent Office in its brief (at 3A-8A).

2. Brown, *Smoothing, Forecasting and Prediction of Discrete Time Series* (1963) (hereafter "Brown") at 101.

In the Brown treatise, the formula is:

$$S_t(x) = \alpha x_t + (1\text{-}\alpha)S_{t\text{-}1}(x)$$

Compare:

$$B_1 = (F)PVL + (1.0\text{-}F)B_0$$

which is Flook's mathematical equation but with the order of terms rearranged so that corresponding terms of the two equations are set one above the other for ease of comparison. $S_t(x)$, like B_1, is the accumulated weighted average at any time t of a variable x whose value changes with time; $S_{t\text{-}1}(x)$, like B_0, is the weighted average at a preceding time $t - 1$; α and $(1 - \alpha)$, like F and $(1.0 - F)$, are the weighting factors; and x_t, like PVL, is the present value of the variable. See Pet. f.n. 3, at 4; and Br. Pet. f.n. 3; at 4; 3A, 6A and f.n. 5.

A related statement was made in 1657 by Christiaan Huygens, a great Dutch scientist, who introduced the important concept of

Thus it is clear that respondent has not invented or discovered any new mathematics or mathematical formulas or equations. The Flook mathematical equation has been and remains in the public domain. A patent to the Flook method would therefore *not* "confer dominion over the equations themselves" (Br. Pet. 10),[3] and the Patent Office's Section 101 objections to the issuance of a patent to Flook no longer exist; See *Kimball v. Kimball,* 174 U. S. 158; *Alton v. Alton,* 347 U. S. 610; *Taylor v. McElroy,* 360 U. S. 610. The factual premise for certiorari was unfounded and moot, and the Question Presented by the Patent Office (Br. Pet. 2) which incorporates that erroneous premise is likewise academic.

Particularly noteworthy is the fact that the entire premise of the Government's Petition for Writ of Certiorari, lies in its assertion of the novelty and innovative character of the mathematical equation. Yet the Government clearly sets forth in its brief its knowledge of the Brown treatise where the same mathematical equation appears.[4] The Government seems to disassociate itself

2. (Cont'd.)

"mathematical expectation"; namely, the sum $ap + bq$ or $ap + b(1 - p)$ where p and q are the probabilities of winning a and b, respectively, and $q = 1 - p$. This concept was presented in the first formal treatise on probability based upon the investigations of the principal mathematicians of that day into the mathematics of gambling. See H. Eves, *An Introduction to the History of Mathematics* (1976), at 288-289.

3. A patent claim for a technological structure (be it an apparatus structure or a process structure) based on a mathematical formula *does not preempt that formula.* To the contrary, the claim is limited to the structure with a precise relationship of parts set by that formula, for "the invention [is] a narrow one" (*Mackay Radio & Tel. Co. v. Radio Corp.*, 306 U. S. 86, 94).

4. The Patent Office formulates a complex "generalized . . . expression . . . of the alarm-base function" not at issue in this case and notes that "Brown . . . deriv[es] a similar expression"

from any responsibility for the determination of novelty by stating that this was the conclusion of the Patent Examiner and that it would "not contest that conclusion" (Br. Pet. f. n. 12, at 14). Thereafter, the Government refers frequently to the "innovative mathematical algorithm" (Br. Pet. 23).

The Government's failure to place before the Court this identical Brown formula in the petition and again failing to do so in the brief for the petitioner are not explained and has resulted in a completely inaccurate presentation in the Petition and in petitioner's brief. A wrong impression has been thereby left that the issue before the Court is that of patentable subject matter under 35 U. S. C. Section 101 (namely, the issue of preempting a so-called "novel" mathematical equation) when in fact the sole issue is whether the technological relationships of an automated oil refinery's alarm-setting method defined by an old mathematical equation are new and unobvious under 35 U. S. C. Section 103.

2. A Section 103 Inquiry Is Implicitly Raised for the First Time Before This Court by the Patent Office.

The Court has established in *Graham v. John Deere Co.*, 383 U. S. 1, 10, the type of "inquiry" to be undertaken under Section 103. That inquiry must be "followed realistically" to determine "'patentable novelty'" (*id.* at 9)

4. (Cont'd.)

(Br. Pet. f.n. 5 at 6A). But nowhere does it simply advise the Court that an equation mathematically identical to Flook's is in Brown. Instead, the Patent Office erroneously speaks of "The only inventive contribution is in the mathematical formula" and "respondent's discovery of the mathematical equations." (Br. Pet. 22 and 23).

when the individual elements of an invention may be old, but their relationship in a combination of these elements forming the invention is alleged to be new as here (Pet. 15, at f.n. 21). The emphasis on nonobviousness of Section 103 furnishes "a practical test of patentability" (*Graham, supra*, at 10).

That "practical test" requires consideration of "the subject matter as a whole" and whether

> as a whole [it] would be obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains (35 U. S. C. 103).

Applied to this case, Flook's inventive contributions can be evaluated for "patentable novelty" only by considering "as a whole" the automated oil refining process with the alarm limits set in accordance with an old mathematical equation.

The form of such an "inquiry" on "nonobviousness" is not changed where an old mathematical formula is employed to achieve a new technological structure. See *Mackay Radio, supra*, 306 U. S. 86, 94, where the Court stated the rule:

> While a scientific truth, or the mathematical expression of it, is not patentable invention, a novel and useful structure created with the aid of knowledge of scientific truth may be.

The test for patentable novelty used by the Court in *Mackay Radio* is similar to that of Section 103, namely "whether it was more than the skill of the art," *id.* at 94, to combine old mathematics and known radio antenna structures to produce a novel radio antenna considered as a whole.

That "practical test" of considering "the subject matter as a whole" (Graham at 10-11) cannot be bypassed for any class of invention. As in *Mackay Radio*, that test must also be applied where mathematical equations are used to build and to define "a novel and useful structure" (at 94) such as the oil refining process achieved by Flook's use of this mathematics (Pet. 15, at f.n. 15).

The Patent Office is asserting that the invention is *obvious* when it argues that the Flook oil refining process is constructed "in conventional ways [and is] entirely predictable" (Br. Pet. 13). However, a Section 103 "inquiry" to determine nonobviousness is required by Congress and approved by this Court in *Graham, supra*, at 10:

> [In Section 103, the] emphasis on nonobviousness is one of *inquiry*, not quality and, as such, comports with the constitutional strictures (emphasis added).

There is only one type of inquiry into "non-obviousness" permitted under the patent statutes, that of Section 103, and the Court in *Graham, supra*, at 11 has spelled out in detail the nature of the procedures to be followed.[5] The Patent Office cannot circumvent that statutory mandate, and dispense with such a Section 103 inquiry under the guise of an examination of patentable subject matter under Section 101.

---

5. "Under § 103, the scope and content of the prior art are to be determined; differences between the prior art and the claims at issue are to be ascertained; and the level of ordinary skill in the pertinent art resolved. Against this background, the obviousness or nonobviousness of the subject matter is determined. Such secondary considerations as commercial success, long felt but unsolved needs, failure of others, etc., might be utilized to give light to the circumstances surrounding the origin of the subject matter sought to be patented. As indicia of obviousness or nonobviousness, these inquiries may have relevancy. See Note, Subtests of 'Nonobviousness,' 112 U. Pa. L. Rev. 1969 (1964)." (*Graham, supra*, at 11).

The Patent Office acknowledges that a Section 103 inquiry in this case was not before the Board of Appeals, "was not before the court below and is not before this Court." (Br. Pet. f.n. 11 at 14; Pet. f.n. 21 at 15). This Court is not equipped to perform *ab initio* a Section 103 inquiry without a record and without the assistance of the tribunals below. Moreover, it would violate the principle that no matter may be argued at the appellate court level which has not previously been raised before the lower court. That principle has particular applicability in this instance where the C. C. P. A.'s "special competence in complex, technical and important matters." [6] would be indispensable.

The Section 103 inquiry implicitly called for by the Patent Office argument is jurisdictionally barred here. 28 U. S. C. 1256 limits the scope of this Court's appellate review to "cases in the C. C. P. A." The section must be read in light of the jurisdictional limitations of 35 U. S. C. § 141, et seq. Because the "obvious" or "predictable" issue was not, and could not have been, in the C. C. P. A. case it cannot be considered by this Court; see § 144.

A Section 103 inquiry is fundamental to the determination of patentability in this case and requires evaluation of the facts of the prior art mathematics never considered by any of the tribunals below. Therefore, the writ of certiorari should be dismissed, and the case returned to the Patent Office, where the Section 103 inquiry can be conducted based on all of the prior art now known.[7]

6. *Glidden Co. v. Zdanok*, 370 U. S. 530, 560.

7. The applicable Patent Office procedure for reopening prosecution after the return of a case from a court decision is described in the Office's Manual of Patent Examining Procedure, Section 1216.01(4) at p. 220.7.

B. Also Erroneously Presented to the Court Is the Question of Patents for Computer Programs or "Computer Applications."

1. The Record in This Case Is Insufficient to Permit Consideration of the Question Presented by the Government.

This Court is being asked by the Patent Office to rule on important factual issues concerning the technology of oil refineries, the automated control of the oil refinery process, as well as computers and computer programs used in the automation. The record below consisting of the patent application and related technical descriptions satisfies the statutory minimum of a disclosure of the invention which would "enable any person skilled in the art . . . to make and use the same" (35 U. S. C. 112). However, it is entirely barren of the type of technological description and background developed at a district court trial through the adversary process which would enable the judge to make his findings of fact and conclusions of law and which would serve as a record for appellate review. The Patent Office has not tendered to this Court or to the court below the kind of additional technological evidence required to support its position, nor could it seek to introduce any such evidence at this state of the proceeding.[8] Since the record "is insufficient to permit decision of [the] claims," [9] the writ of certiorari should be dismissed, see *Massachusetts v. Painten*, 389 U. S. 560.

Some of the major technical fact issues raised by the legal contentions include

1) What is the oil refinery process considered "as a whole" (Section 103); what are its parts that produce the

8. See 35 U. S. C. § 144, which limits the C. C. P. A. to deciding the appeal "on the evidence produced before the Patent Office."

9. *Johnson v. Massachusetts*, 390 U. S. 511.

"process variables" (patent claim 1) whose values are employed in setting the alarm limits? What is an automated oil refinery and why is an alarm required? How is an alarm setting control integrated into an automated system and how are computers and computer programs structured for this process? How does the oil refinery work without Flook's automated alarm setting, and how does it work with it?

2) Is the programmed computer a "concrete embodiment in a specifically devised apparatus" (Pet. Br. 14)? Does programming a general-purpose digital computer restructure the computer into a new machine with a different structure? The Patent Board of Appeals in *Ex parte Johnston* [10] found that it does,[11] and the C. C. P. A. has consistently and unanimously applied that technological principle from the earliest cases.[12] The petitioner seems to suggest that he opposes the restructuring principle (Br. Pet. 14) but tenders no technological evidence to support such a position. In the absence of a factual record of this technology, it is completely inappropriate for the Patent Office to request a rule that would proscribe "the patenting of computer applications." (Br. Pet. 23) and deprive the software industry of the benefits of the patent system.

3) Are hardware- and software-built embodiments of the same machine technologically equivalent? The Patent Office requests a ruling that a software-built machine-system or process is unpatentable solely because it is built in software (see Pet. Br. f.n. 24, 25, at 24). Such a ruling would be completely at odds with the well-established

10. Reported in *Dann v. Johnston* 74-1033, Petition for Writ of Certiorari, 25A, at 28A.

11. See also *infra* at p. 17 for other findings by the Board.

12. *In re Prater and Wei*, 415 F. 2d 1393 (1969); *In re Bernhart*, 417 F. 2d 1395 (1969).

technological "principle of the equivalence of [software] programming and hardware. This principle [is based on the fact] that a program for a general-purpose machine turns it into a special-purpose machine." [13] The Patent Office offers no evidence to the contrary, nor can it justify the different patent-law treatment that it proposes for software-embodied inventions from that accorded hardware-embodied inventions.

This is the fifth computer programming case [14] that the Patent Office has brought to the Court by way of certiorari. It is also the fifth time that the Patent Office asks for a sweeping ruling against patents for machines and machine processes that are constructed with software technology. As in all the preceding cases, this case is devoid of a technological fact record that would justify such a ruling. The Government has not been able—neither with its own resources nor with those of the CBEMA hardware manufacturers—to supply the Court with such a record of technological facts. There is only one conclusion: there are no such facts.

2. Improperly Presented to the Court Is the Request For a Rule Declaring Machines or Processes Built With Computer Programs to be Unpatentable Subject Matter.

This case was presented to the Court as a vehicle for a ruling that software-built special-purpose computers are unpatentable subject matter. However, this Court cannot issue such a rule of unpatentable subject matter under

13. Prof. Saul Gorn, Final Report No. AD 59 URI, at p. 4, from Univ. of Penna. Moore School of Elec. Eng. to U. S. Army Signal Research & Development Labs, 1959.

14. The others are *Gottschalk v. Benson*, 409 U. S. 63; *Dann v. Johnston*, 425 U. S. 219; *Dann v. Chatfield*, No. 76-1559 cert. dnd. — U. S. —; *Dann v. Noll*, No. 76-1558, cert. dnd., — U. S. —.

§ 101, based on the way a machine is built, without violating the spirit and letter of the patent statute. Whether a particular machine constitutes patentable subject matter depends on the well-established principles of patent law—not whether the machine is built with one form of construction (software) as against another (hardware).

This Court cannot rule that all programmed general-purpose computers are unpatentable any more than it could rule a radio built with vacuum tubes is patentable, but if built with transistors, it is unpatentable. The patent statute is broadly drafted, defining as patentable all machines and technological processes using machines regardless of the technology by which they are built.[15]

Unable to decide the principal issue for which certiorari was sought and granted, this Court should exercise its discretion to decline to review the case and dismiss the writ of certiorari as improvidently granted.[16]

II. Special-Purpose Computers Built by Programming a General-Purpose Computer Form Machines and Machine Processes, Hence Patentable Subject Matter.

A. The Technology of Computers and Computer Programs.

The technological facts of the nature of programmed computers are made clear by (a) the computer experts,

15. The only exception of a particular technological field, that of atomic weapons, is explicitly defined by Congress, 42 U. S. C. § 281(a).

16. The dismissal of certiorari would not leave the C. C. P. A. decision as a final adjudication that respondent's invention is patentable subject matter. The validity of any patent that may issue is always subject to attack in an inter partes proceeding (see Reviser's note to 35 U. S. C. § 144) where a full record could be developed, see *Glidden*, *supra* at 578-79. Moreover, the Patent Office, upon remand, may undertake a Section 103 inquiry based on its having learned in the course of these certtiorari proceedings that the mathematical equation is not new.

(b) the C. C. P. A. findings in other cases, (c) the Patent Office's own findings some years ago, and (d) the Board of Appeals' specific fact finding in the *Johnston Case*.[17] The computer experts explain the facts of the well-known "feature distinguishing the computer from other mechanisms . . . its restructurability. . . . The program restructures the computer."[18] The findings of fact as to this computer technology by the C. C. P. A. in a long line of cases up to the present[19] have held that a general-purpose computer is a "storehouse of parts":

> But once a program has been introduced, the general-purpose digital computer becomes a special-purpose digital computer (i.e., a specific electrical circuit. . . .).[20]

The special-purpose computer "is physically different from the machine without the program" and is patentable as a machine defined as a new combination of elements under 35 U. S. C. § 112.[21] The Patent Office in its Guidelines for Examination of Computer Programs (1966)[22] written in

17. See f.n. 10, *supra*.

18. Professor Ivan Flores, City University of New York "Computer Software," *Science & Technology*, No. 89, May, 1969, pp. 16, 17.

19. *In re Prater and Wei*, 415 F. 2d 1393 (1969); *In re Bernhart*, 417 F. 2d 1395 (1969); *In re Comstock*, 481 F. 2d 905 (1973); *In re Knowlton*, 481 F. 2d 1357 (1973).

20. *In re Prater*, *supra*, f.n. 29.

21. *In re Bernhart*, *supra*, at 1400, Judge Rich in the *Johnston Case*, 502 F. 2d 765, 773 reaffirmed "the validity of this principle [namely] a new program makes an old general purpose digital computer into a new and different machine." There has always been unanimity and no dissent in the court below on "the validity of this principle."

22. 829 O. G. 441, 442, by Order of Commissioner of Patents Brenner, August 9, 1966.

part by members of the Patent Office Board of Appeals,[23] recognized that the stored program converts that "'warehouse' of unrelated parts" of the general-purpose computer into a "special-purpose machine" which may be patented as "a novel and useful machine combination due to the interrelation of parts . . . and the fact that portions of the complete machine take the form of a replaceable program is of no moment."

In the *Johnston Case,* the Patent Board of Appeals found:

> . . . that an unprogrammed, or a differently programmed, general-purpose computer is [not] the same machine as the one that would result from appellant's programming of the same or of a similar computer.[24]

The following discussion of the nature of the computer and stored-program technology is presented in order to assist the Court in understanding that respondent's preferred embodiment of a programmed computer for automating an oil refinery's alarm limit control is a "concrete embodiment in a specially devised apparatus," the Patent Office's contrary and unsupported assertion (Br. Pet. 14) notwithstanding.[25]

23. *See* Commissioner of Patents Brenner, "The Future of Computer Programs in the United States Patent Office," *Law of Software,* 1968, p. B-8.

24. Reported in *Dann v. Johnston,* No. 74-1033, Petition for Writ of Certiorari, 25A, at 28A.

25. Petitioner's attempt to have this Court determine by legal fiat and contrary to the facts, the nature of computers and computer programs brings to mind the declaration of the Court of the Special Inquisition that the sun rotated around the earth in the face of Galileo's evidence of a different structural relation of sun and earth; *see* J. Bronowski, *The Ascent of Man,* (1973) p. 213, *et seq.*

1. Respondent's Programmed Computer for Automating an Oil Refinery Is a Circuit Configuration.

Respondent's preferred embodiment of a programmed computer is constructed as a configuration of electrical components formed in "a specific electrical circuit".[26] This structure of respondent's machine (and of other machines implemented with a general-purpose digital computer and an internally-stored computer program) is illustrated in the accompanying diagram of the hierarchical structure of a special-purpose computer (see p. 20).

In this diagram, each level of the hierarchy,[27] supplies the building blocks out of which the circuits of the next level of the system[28] are built. The bottom level consists of electrical components from which the logic circuits are combined into the more complex building blocks —circuits called the "functional units"[29] of the general-purpose computer's processor.[30] These functional units, in turn, are combined into the special-purpose computer (such as respondent's) which receives input signals (for the values of the oil refinery's "process variables") and pro-

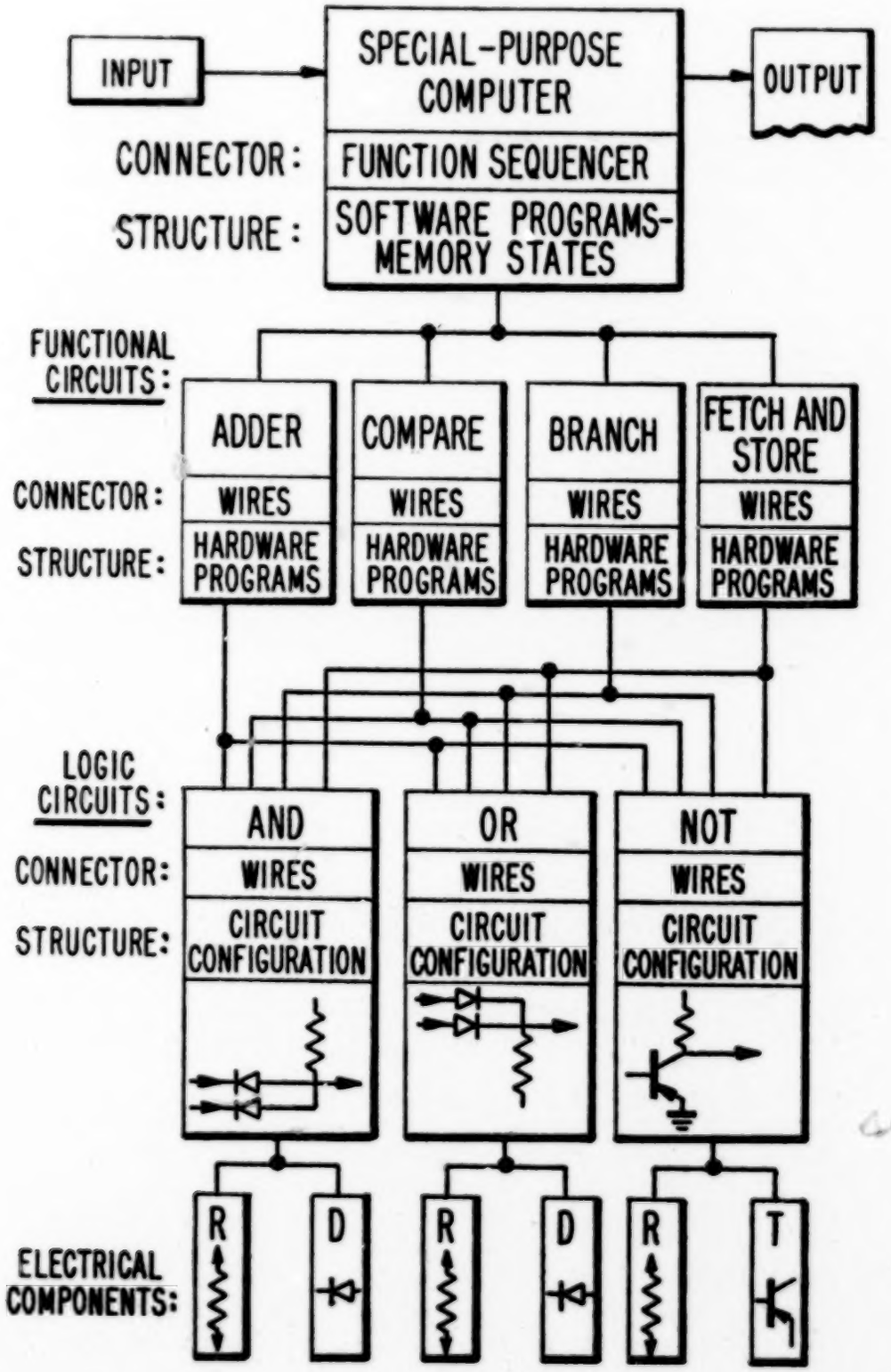
26. *In re Prater, supra,* f.n. 29.

27. "*Hierarchy* . . . a rank arrangement in which the nature of function at each higher level becomes more broadly embracive than at the lower level." (A. Iberall, *Toward a General Science of Viable Systems,* McGraw-Hill, 1972, p. 388.

28. "*System,* A complex unity formed of many, often diverse, parts subject to a common plan or serving a common purpose. . . ." (Ibid. at p. 7).

29. Professor Ivan Flores, *Computer Logic: The Functional Design of Digital Computers,* (W. L. Everitt, Editor) Prentice-Hall, Inc., 1960, Chapter Eleven, "Functional Units," pp. 173-207.

30. And into the other circuits (*e.g.,* memory, sequencer control, input and output) that form the general-purpose computer. "The term *logic* . . . describes the interrelation among the primary building blocks of the computer." Flores, *ibid.,* p. 123.



duces output signals (the updated alarm values used to adjust the refinery's alarm limit).

The electrical components, e.g., resistors (R), diodes (D), transistors (T), at the bottom level [31] of the hierarchy are like those used in all modern electronic equipment (e.g., in radar, television, radio, telephone, automation). These electrical components are combined in different circuit configurations to form logic circuits called AND, OR and NOT which transform electrical signals in a manner representative of their named logic operation.[32]

The logic circuits are structured at the next level into functional circuits (Adder, Compare, Fetch and Store, and Branch or Conditional Transfer of Control). These functional circuits combine electrical signals in accordance with their named function. The Adder functions to add numbers represented by combinations of signals. Similarly, the Compare circuit operates to compare numbers for the relations of "equality" and "greater than" and "less than." Fetch and Store are control circuits for moving data signals to and from the memory, which is constructed from millions of flip-flop circuits, the basic storage units for the binary computer signals.[33]

The Branch or Conditional Transfer of Control performs a function unique to digital computers, whereby the sequence of functional circuits can be varied in accordance with different *conditions* of the input or processed

31. At a still lower level of this structural hierarchy, the molecular and atomic substructure of these components (such as the solid-state transistors and diodes) are important factors in their design and in their interrelationships at higher hierarchical levels.

32. The AND circuit produces an active output signal when all of the circuit's inputs are active; OR, when either one or another, or all of its inputs are active; NOT produces an active output when its input is not active. In place of three different types of logic circuits, many digital computers are built from combinations of a single such circuit, the NOR (the negative of OR; its output is active when neither of its inputs are active).

33. The flip-flop is also formed from the logic circuits.

data. In a simple example, different sequences of circuits are activated depending on which of two numbers are activated.[34]

A general-purpose computer is configured to have a basic "repertoire" of functional circuits (also "instructions" or "orders").[35] The structural configuration of each of these functional, control and memory circuits is called "permanent programming"[36] or "hardware programming."[37]

Programs in the broader sense are part of all automatic machines, and automated systems. "*Program elements* determine *what* the automated system shall do and *how* the parts of the system must function in order to accomplish the desired result."[38]

"Software programming"[39] is the structure of the special-purpose computer which configures the functional circuits (Adder, Compare, Branch, etc.) to interconnect certain sequences of those circuits and combinations of those sequences.[40] The software programming (the internally stored program) takes the physical form of combinations of different states of the memory circuits and of electrical signals that control the states and configurations of the functional circuits.

34. This Branch control (or "Decision") is fundamental to the ability to structure a general-purpose computer into a special-purpose computer by means of software programming.

35. A more elaborate set (*e.g.*, to perform also subtraction, multiplication and division, shifting of numbers, etc.) is often used.

36. I. Flores (1960), *supra*, p. 42.

37. Prof. Saul Rosen, "Hardware Design Reflecting Software Requirements," 33 *Proc. Fall Joint Computer Conference*, 1968, Pt. II, pp. 1443, 1444.

38. "The program may consist of mechanical parts such as gears, cams, and linkages that connect the various portions of a machine together and determine the timing of the various actions of the machine. In the most modern computer-controlled systems, the program may be stored in the form of punched paper cards, paper tape, magnetic tape, ferrite cores, electronic circuits, or other mechanical or electronic "memory" devices. 2 *New Encylopedia Britannica: Macropaedia*, 505, 506, "Automation."

39. Term used herein for what is customarily called "computer programming" to avoid semantic confusion with hardware programming.

40. Basic repetitive sequences are called "sub-routines." (Br. Pet. 15).

Software programming resembles hardware programming in a number of respects, and most importantly in the way that lower-level building blocks are structured.[41] However, the connector used with the software for configuring the functional units is a different type from that of the hardware programming. Starting at the bottom level of the hierarchy, as shown in the diagram, the electrical components are connected together in the logic circuits by fixed wire or "hardwire" connectors.[42] The logic circuits are similarly hardwire connected into the functional circuits and other circuits of the general-purpose computer. However, at the top of the hierarchy, a "softwire" connector is employed for the software structure. This connector includes a function ("instruction") sequencer, which effectively interconnects the functional circuits in the sequences prescribed by the software programming.[43]

By forming the structure at the top of the hierarchy as electrical signals, that structure can be stored in the computer memory in the same way that other signals are

41. The software and hardware programming are engineering equivalents, as discussed hereinafter.

42. "Hardwire" connections may be made by simply twisting or soldering wires together or in more advanced fabrication by printed circuits or integrated circuits; *see Scientific American* article, May, 1975, "Microprocessors."

43. The function sequencer activates by control signals the circuitry of the functional units at the appropriate times in sequences specified by the software programming. When activated the circuits are effectively "connected," and when inactive they are not. The engineer's term "softwire" refers to use of control signals for the connections. The sequencer, memory, input and output circuits are all part of the general-purpose computer; the control program is in the states of the memory circuits.

stored. The ability to store the structure of a special-purpose computer in the form of electrical signals gives rise to the ability to replace one such structure by another simply by replacing one set of electrical signals (software program) by another. *This concept of the stored-program computer is the foundation on which the computer industry is built.*

Since the building blocks of each hierarchical level of such computers are nested within those of the next higher level as shown in the diagram, the special-purpose computer at the top level includes all of the elements and structural relations shown in the drawing. This special-purpose computer receives input signals and, within its circuits, combines them and modifies them to produce output signals. The combining and modifying is in accordance with the structure of the stored software programming, which uniquely establishes the "rule of action" [44] of the special-purpose computer.

Respondent's special-purpose computer, in the same way, functions as "a specific electrical circuit" [45] configured out of sequences of functional circuits, which in turn are configured combinations of logic circuits, which in turn are configured combinations of circuit components.

2. The General-Purpose Computer Is an Incomplete Machine That Serves as "a Storehouse of Parts" and Is Completed by Respondent's Stored Program to Form a New Combination of Circuits as Respondent's Programmed Computer for Automating an Oil Refinery.

The general-purpose computer, though extremely complex, is an inchoate, and *incomplete* machine. The functional circuits (Adder, Compare, Branch, etc.) are

44. Robinson on Patents, Vol. I, p. 257, § 173.
45. *In re Prater, supra.*

not interrelated nor do they co-act in any useful way until structured [46] by the software programming into a special-purpose computer. The functional circuits are building blocks used by the software; and in that respect the general purpose computer is a "'warehouse' of unrelated parts." [47]

This "storehouse of parts" [48] character of the general-purpose computer is evident from the hierarchical diagram on p. 21. Starting at the bottom level, each level serves as a storehouse of parts for the next higher level. The electrical components (resistors, diodes, transistors) serve no purpose by themselves and acquire functions only when connected in the various circuit configurations of the logic circuits. Similarly, the logic circuits serve only as a storehouse of parts for assembling the different circuit configurations of the functional units, and the latter circuits which constitute the general-purpose computer are interconnected by the software programming in the various sequences and combinations that make up the special-purpose computer.

The set of building blocks at each level is similar to the familiar toy Erector Set, whose parts can be connected in an unlimited number of ways to form machines of a wide variety of types. For example, with a storehouse of the electric components (resistors, diodes and transistors, etc.) almost all of the known electrical circuits can be built, including those of the computer logic circuits. With a storehouse of logic circuits, all of the various functional and other circuits of general-purpose computers can be built.[49] Similarly, the function circuits are interconnected

46. *See* Flores, (1969), *supra*, pp. 16, 17.
47. Patent Office Guidelines (1966), *supra.*
48. *In re Prater, supra.*
49. Special-purpose computers are also built directly from the logic circuits using permanent or hardware programming. One could also use the logic circuits as the functional units of the general-purpose computer, namely, the building blocks for software

by the software into an endless variety of automatic machines—a musical synthesizer, the controls of a lunar landing module, a document reader for electric meter reports, a calculator for thereafter calculating and printing the associated electric bills. Respondent's programmed computer is similarly constructed in its preferred form of construction.

There is no predetermined relationship of the function circuits (Adder, Compare, Branch, etc. or their individual functions) to any of the above special-purpose computers and other devices; they are "*unrelated* parts" [50] (emphasis added). Nor is there a predetermined relationship of the resistors, diodes and transistors to a computer or television set. However, when the functional circuits of the general-purpose computer are structured by software programming—such as that of any of the above examples—these circuits are organized and interrelated by that programming just as are the electrical components in a television circuit. In that condition, the function sequencer makes the softwire connections called for by the stored program so that the functional circuits are connected into a special-purpose computer having the structure called for by the program.

The hardwire programming is used in building the functional circuits from the logic circuits and softwire programming is used in inter connecting the functional circuits into a special-purpose computer; both connect building blocks into more complex structures.

In short, as the Government asserted in another case, a software program may be "a technologically innovative

49. (Cont'd.)

programming; but "such an approach would be impractically slow." Goldstine, H. H., *The Computer From Pascal to von Neumann*, (1972), p. 257.

50. Patent Office Guidelines (1966), *supra*.

product" [51] and may be demonstrated to have "technological superiority".[52]

The phrase "patents for computer programs" as used by the Patent Office to describe the key issue in these "computer programming" cases,[53] is technically imprecise and misleading in that what is sought to be protected in this technology is not some abstraction unrelated to a machine, but the structure of a particular special-purpose machine or machine-control process which may be produced by the softwired or programmed general-purpose computer but which also may be hardwired.

3. Hardware and Software Programming are Equivalent and Alternative Approaches for the Construction of Computerized Control Systems and Equally Patentable.

The Patent Office has set up a false dichotomy in the patentability of computerized control systems. Those implemented by hardware programming (*e.g.*, the general-purpose digital computer) are recognized to be machines by petitioner and patents covering such machines and their control processes are granted subject to their meeting the statutory tests of novelty (35 U. S. C. § 102) and unobviousness (35 U. S. C. § 103). However, computers implemented by software programming are generally denied patentability.[54]

51. Government's Pretrial Brief in *United States v. IBM*, 69 Civ. 200, at p. 202.

52. *Ibid.* at p. 29. This Government Brief in the IBM case makes it clear that the software industry is in the business of selling technological products (machines) not "abstract ideas."

53. *Benson, Johnston, Chatfield,* and *Noll* (see f.n. 14, *supra*), the predecessor cases.

54. The Patent Office is not consistent in this respect; many of the Examiners understand the machine structure of software, as shown by the patent for an "Analytic Character Recognition System" implemented by software, U. S. 3,873,972.

Petitioner's distinction between the two alternative forms of computer construction is inconsistent with the well-established technological "principle of the equivalence of [software] programming and hardware. This principle [is based on the fact] that a program for a general-purpose machine turns it into a special-purpose machine." [55]

Petitioner has acknowledged in the previous *Johnston Case* this equivalence of hardware and software, namely, that a special-purpose computer can be built either by hard-wired logic circuitry or by programming a general-purpose computer.[56] The particular means that is used depends upon many business and engineering trade-offs.[57]

55. "The principle becomes more tangible when we compare in detail the logical design of that special-purpose machine with the flow chart of that program." Prof. Saul Gorn, Final Report No. AD 59 URI, at p. 4, from Univ. of Penna. Moore School of Elec. Eng. to U. S. Army Signal Research & Development Labs, 1959.

56. "The particular operations to be performed by a computer are governed by the state and configuration of its electronic circuitry [which] can be fully built into the computer through wires. . . . Alternatively, the computer may be designed so that the state or configuration of its circuitry can readily be changed without rewiring or other visible physical changes . . . by use of different 'programs.'" (See *Dann v. Johnston* No. 74-1033, Brief for the Petitioner at 14-15.)

57. "Perhaps the most important part of the job of computer system design is to determine which functions are best handled by hardware, and which should be provided by software. All of the traditional engineering considerations come into play here. One must consider the added cost of additional hardware measured against possible gains in operating efficiency. One must consider the availability of manpower with the appropriate skills for design and for implementation. One must also consider intangible factors such as the inherent advantages of simplicity and the hidden costs of complexity.

"From the point of view of the computer user it makes no difference which features are incorporated into the hardware and which are programmed in software.

"It is the combined hardware-software system, the 'extended machine' that the user must rely on to provide the power and flexibility and convenience that have been promised with each new generation of computing equipment." S. Rosen (1968), *supra.*

None of the factors concerning the choice of hard-or-softwiring has any relation to inventiveness of the underlying machine. If the circuit states and configuration of a particular special-purpose computer contain the elements of invention, the machine is patentable and the fact that the wiring is "hard" or "soft" is immaterial, for the two types of computer construction are interchangeable.

This equivalence is inherent in the concept of the stored-program computer. The first electronic computer, the ENIAC built about 1945, was "not a stored-program computer. Programs were installed and changed by engineers who changed the wiring among its various components." [58] The stored-program concept, invented while the ENIAC was being built, had as its very purpose the replacement of the hardwire connections by the softwire. The first published description [59] of the stored-program computer clearly explained (p. 1, ¶ 1.2) that the program ("instructions" [60]) was "an integral part of the [computer's] structure."

58. S. Rosen, Purdue University, "Electronic Computers: A Historical Survey," *Computing Surveys*, Vol. 1 (1969), pp. 7, 8.

59. A. W. Burks, H. H. Goldstine, J. von Neumann, *Preliminary Discussion of the Logical Design of an Electronic Computing Instrument*, 1946. In this famous "paper that became a landmark in the history of computer sciences if not in the history of human thought," (4 *New Encyclopedia Britannica: Macropaedia*, 1045, 1047) John von Neumann and his colleagues at the Institute for Advanced Study in Princeton first published "this stored-program concept" on which the modern computer is based and the technology of which underlies the issues in this case.

60. A human receives *instructions* on how to operate the manual choke in an automobile. When an *automatic* choke is built in by means of a "program," we don't think of "instructing" the engine or its controls. Yet that is the terminology of the computer industry established thirty years or more ago to describe the automatic controls of a stored program. The automatic control of the choke in future cars will be done by a more advanced programmed general-purpose computer, and appropriate computer "instructions" will be used for that purpose. *See* Vacroux, *infra*, f.n. 76, at p. 40.

As far back as 1949, when the first stored-program computer was constructed in England, its builders recognized the equivalence of programming by hardwire and softwire, and explained that additional functional circuits could be added to the computer or the equivalent software programming could be used to implement the same functions.[61]

The Patent Office has no difficulty in finding patentable subject matter in the hardware approach to building special-purpose computers which technologically began to be replaced 30 years ago with stored-program inventions. However, it insists that the stored program approach which is the present foundation of the computer industry and is revolutionizing the entire electronics industry is not patentable. To base disparate legal treatment upon a mode of construction has no justification in law, technology or common sense. Softwired (*i.e.*, wireless) programming is an historic advance over the wired program. The patent statutes and their constitutional base were not intended to be limited to a finite period of technological development. There is no justification for the Patent Office's position that technology has now outstripped the boundaries of patent law.

The equivalence of hardware and software underlies this revolutionary change taking place today in the electronics industry. Programmed microcomputers (general-purpose computers constructed with "microscopic components"[62]) are used to build electronic apparatus in "a wide range of applications from specialized laboratory instruments to sales terminals and electronic games."[63]

61. M. V. Wilkes, "Progress in High-Speed Calculating Machine Design," *Nature*, Vol. 164 (1949) pp. 341-3.

62. A. G. Vacroux, "Microcomputers," *supra*, pp. 32, 33.

63. *Ibid.*, p. 34.

"Until recently digital circuits were designed almost exclusively with 'hardwired' logic . . . although this approach is very powerful (indeed, it is used in most of the electronic equipment manufactured today), it yields an *inflexible* product that is difficult to modify or improve without major redesign. Most of the microcomputers in service today were adopted as a replacement for *inflexible* circuitry. . . . *it is to their flexibility, inherent in all stored-program systems, that they owe most of their popularity.*[64]

Thus, the general-purpose computer has become (in miniature size) a tool that is structured by the addition of programs to build all kinds of instruments and apparatus. Each programmed microcomputer is a different electronic machine as is each hardwired device that the programmed microcomputer is replacing. Such electronic machines have unquestionably been patentable subject matter under 35 U. S. C. § 101 when hardware programming was used. The objection to patentability based upon the use of the more modern and "flexible" technology of the stored program cannot be justified by any basis in law or fact.[65]

The denial of patents to a modern technology (the softwire mode of building special-purpose computers and other machines) raises anew the question: whether the

64. *Ibid.*, p. 39.

65. The failure to recognize the equivalence of hardware and software opens a pandora's box and leads inevitably to a "raiding" situation in which existing hardware patents have no legal protection from infringing software-constructed devices. *See Digitronics Corp. v. The New York Racing Association*, (E. D. N. Y.), 187 U. S. P. Q. 602, affd. on other grounds 553 F. 2d 740, cert. dnd., — U. S. —. This problem is not limited to the computer field, for many automatic machines have the potential to employ a programmed general-purpose computer to replace their existing electronic, electrical or mechanical controls. *See* 2 *New Encyclopedia Britannica, "Automation," supra.*

Patent Office "freeze[s] . . . patents to old technologies, leaving no room for the revelations of the new, onrushing technology." (*Benson, supra,* at 71.)

4. Software Is Not an "Abstract Idea" and Its Construction Is Engineering.

The software products segment of the computer industry is composed of hundreds of small and medium size companies, many of which are members of ADAPSO. The programmers build software out of building blocks to engineer the special-purpose computer in its hierarchical structure,[66] much the same as that required for other modern complex machines.

In the hierarchy of computer systems, the logic circuit designer builds logic circuits such as "AND" and "OR" circuits without concern for the molecular structure of the individual transistors and other components that make up those electronic circuits (which concern is left to component designers working at a lower level of hierarchy). The designer of the hardware function circuits such as "ADD," "COMPARE" and "STORE," configures logic circuits without concern for the detailed structure of the individual logic circuits he uses to construct the hardware function circuits. Similarly, the "programmer," the designer of the special-purpose computer, configures hardware function circuits into the structure of the ultimate special-purpose machine without concern for the detailed structure of the individual hardware function circuits. The same hierarchical process is followed if, instead of softwire programming, hardwire programming is used to build the special-purpose computer.

66. See "Hierarchies—Software Engineering Design," Software Engineering, International Computer State of the Art Report, 1972, p. 47 et seq.

These building blocks lie on successive levels of a hierarchy of circuit construction and the software engineering is performed at the highest level. The software itself, because of its complexity, often must also be constructed in a number of hierarchical levels, and the software engineer should preferably work like:

> ". . . the electrical engineer designing a new T.V. set [who] certainly doesn't design each vacuum tube, transistor, and capacitor all over again; he relies on existing components. In fact, he normally designs on a much higher level, using 'off-the-shelf' power supplies, oscillators, etc." [67]

A recent IBM report further demonstrates the similarities in software and hardware engineering:

> Structured programming reduces the arrangement of the program logic to a process like that found in [hardware] engineering where logic circuits are constructed from a basic set of figures. As such, it represents a standard based on a solid theoretical foundation.[68]

Working at different levels in the system hierarchy of the special-purpose computer to be built, contemporary software engineers as that IBM report explains, may use a "structured programming" [69] approach similar to the design of other automatic machines:

> The top-down approach is patterned after the natural approach to system design and requires that

67. The IBM report, "Composite Design: The Design of Modular Programs," Jan. 1973, p. 6.

68. "Management Concepts for Top-Down Structured Programming," IBM Report, February 1973, p. 1.

69. *Ibid.*, Abstract.

programming proceed from developing the control architecture (interface) statements and initial data definitions downward to developing and integrating the functional units.

In the Apollo Space Shuttle Software System, a mammoth software development effort:

> . . . program structuring is of major importance; decisions . . . determine the total structure of the software both statically and dynamically. Program structuring defines the building blocks, the control mechanisms and the interfaces of the software. It is in this effort that the interrelationship of systems software, applications software and data is entirely mapped out.[70]

The software engineer builds complex automatic machines as the foregoing examples amply demonstrate. Neither the software engineer nor the hardware engineer devises "abstract ideas" or "mathematical algorithms," contrary to the Patent Office's suggestions (Br. Pet. 17) and to what CBEMA would have the Court believe (Br. CBEMA 17).

The design and development process in this technology is quite similar to that of any other technology. First the goal is decided upon, then the prior art is explored for possible solutions and, if not found, the "technological roadblocks" are identified.[71] Invention may take place in overcoming these roadblocks.

Parallels to the design and development process of other machines may be seen in the software design process

70. "Top-Down, Bottom-Up, Structured Programming & Program Structure," Draper Labs, MIT, Dec., 1972, pp. 28, 31.

71. *See, e.g., Computers in the 1980's*, R. Turn, Columbia Univ. Press, 1974, p. 21.

presented in a paper at the World NATO Conference entitled "Software Engineering" [72] which took place in Germany in 1969 and was attended by over 50 experts and prominent leaders in the software field.

B. Copyrights Cannot Protect Machine Inventions; Patents, Which Can, are Needed for Software.

CBEMA's suggestions that copyright is a viable method of protecting inventions in programmed computers is contradicted by the constitutional limitation of copyright protection to "writings." Copyrights may only be used to protect particular printed records and descriptions of machines, but not the actual machine structure that they represent. To one whose innovation is a machine structure for a special-purpose computer built with software, only patent protection will suffice.

The copyright laws contain no novelty requirement; exclusive rights are predicated minimally on originality. CBEMA favors copyright protection because its members apparently are not interested in protecting the innovation in software (which may exist in 1% of all programs) but are interested in protecting the non-innovative aspects of software (perhaps 99% of all programs), the part of a machine which has never been protectable.

III. RULE OF BENSON.

The Court, in *Dann v. Johnston*, 425 U. S. 219 (after extensive briefing on the patentability of software-embodied machines, and on the technology of general-purpose computers, computer programs and programmed general

72. "Software Engineering," Report on a World Conference Sponsored by the NATO Science Committee, Garmisch, Germany, Jan., 1969, p. 13.

purpose computers) reaffirmed that the rule in *Gottschalk v. Benson*, 409 U. S. 63, was "a limited holding"; namely:

> "[t]he [Benson] claims were not limited to any particular art or technology, to any particular apparatus or machinery, or to any particular end use." [409 U. S.] at 64. . . . Our limited holding, id., at 71, was that respondent's method was not a patentable "process" as the term is defined in 35 U. S. C. § 100(b).

The Patent Office had sought in *Johnston* to extend the *Benson* rule to deny patentability to machines and machine processes constructed with software programming. But the Court's response was to emphasize the *limited Benson* holding. The Patent Office now seeks to resurrect the matters laid to rest in *Johnston*.

In *Benson*, the absence of claim limitations "to any particular art or technology, to any particular apparatus or machinery, or to any particular end use" made it clear that the claim was for an abstract mathematical process that cannot be preempted by a patent. *Benson* did not find non-patentability because software (rather than hardware) was the mode of construction.

Respondent's invention is directed to an innovation in the automated controls of an oil refinery, and particularly to the setting of the control limits for an alarm needed in the automated oil refinery. Unlike *Benson*, the claims herein are limited to a "particular art and technology . . . and to a particular end use," that of the oil refining process and the adjustment of the refinery's alarm limits. The claims are also limited to "particular apparatus or machinery" involved in the oil refinery process and particularly that of the alarm and its control limits. The only thing that would be patented by these claims is a technological process.

The computer technology deals with a subhuman activity that superficially is associated with the mind.[73] But any patent to respondent's automated oil refinery process controlled in the preferred embodiment by his special-purpose computer would not cover any "abstract idea," because any oil refinery operator is free to practice the mathematics except in the process of setting its alarm limit control. The latter technological process is no more the embodiment of an "abstract idea," than any other part of the oil refinery process.

IV. NATURE OF THE SOFTWARE PRODUCTS INDUSTRY.

A. The Delayed Development of Competition in Software.

ADAPSO is an organization of relatively small companies. CBEMA is an organization of relatively large companies led by IBM [74] in the computer hardware industry but including such giant companies as Sperry-Rand, Honeywell, Burroughs, Control Data and National Cash Register.[75]

Most of the independent software package suppliers (software products manufacturers) are small firms employing fewer than 25 people and having less than $1 million per year in revenues. One half of the suppliers

73. "[I]n the words of James Clerk Maxwell, 'the human mind is seldom satisfied, and is certainly never exercising its highest functions, when it is doing the work of a calculating machine' . . . Just as the use of man's muscles for pumping water was inefficient, so was the use of his mind for carrying out calculations . . . Computing is thus subhuman in that it calls on very few of man's manifold abilities and yet is fundamental to many of his other activities." *Goldstine, supra*, at p. 343.

74. Government's Pretrial Brief, *U. S. v. IBM*, 69 Civ. 200, S. D. N. Y. (hereinafter Pretrial Br.).

75. See Report, Brooks Subcommittee on H. R. 4845, Aug. 17, 1965, 89th Cong. 1st. Sess. (House Rep. No. 802) "Domination of ADP Industry by IBM," p. 11.

have been in the software package business five years or less. Only 11% of the software package suppliers are publicly financed.[76]

CBEMA distorts the history of the computer industry when it asserts that "the computer industry has developed in an atmosphere of free interchange of computer programs. . . ." [77] How "free" the atmosphere has been in the computer industry is best shown by the Government's pretrial brief in *United States of America v. International Business Machines Corporation.*[78]

The Government in its antitrust action against IBM notes the development of the computer industry and finds that in "the earliest days of the development of the computer industry, there was nothing that could be identified as a software industry." [79] This was in part because users

76. Frost & Sullivan Computer Software & Packaged Services Market, Dec. 1977, p. 51.

77. CBEMA Brief 17-19.

78. Pretrial Br. *supra.*

See also Berteau, "IBM and the Structure of the Industry, The Software Monopoly", 21 *Datamation* 111, October, 1975:

> We have seen hardware technology continue to advance at a rapid pace but the systems architecture seems to stay the same. Have you wondered why? The answer is *software.* As long as the structure of system software is fixed, innovation in system architecture is limited. . . .
>
> The most significant reason for this situation probably lies in the history of software building by manufacturers. It's just not in the manufacturers' interest to produce optimum system software. It might result in the manufacturer selling less iron. . . .
>
> The solution to the software building problem brings us to a real issue facing the industry today—monopoly and restraint of trade in software. We have a small but significant competitive market in applications packages. There is very little competition in sorts and compilers, almost no competition in operating systems. And unless the U. S. antitrust lawyers understand the significance of this, we'll have even less competition in the future.

The author is identified by the magazine as "a holder of one of the early software patents."

79. Pretrial Br. 28.

design some of their own software and because the "computer systems manufacturers, following IBM's lead, provided [free] much of the software [that] users required." [80]

It was not until the late 1960's that "the independent software *products* market emerged. As the Government notes, "in light of the fact that IBM provided its customers with their software requirements as part of the IBM 'total systems solution,' independent software product firms had to market their offerings against products for which there was no identifiable price. As a result, only those companies which could offer products not available 'free' from IBM or which could demonstrate the technological superiority of their products in comparison to IBM's had any chance of survival." [81] The result was that the independent software product firms lacked "a realistic opportunity to compete." [82]

The Government has also noted that "IBM internal memoranda . . . reflect that from as early as 1965, IBM was aware of the marketing benefits derived from bundled pricing and the anticompetitive effect it had on software companies." [83] The net effect was that "bundled pricing enabled IBM to exclude competition from independent software vendors." [84] The limited unbundling that began in 1970 when IBM began pricing some of its software has made competition more possible. In 1977, revenues of independent suppliers of software packages rose to $770 million.[85]

80. Pretrial Br. 28.

81. Pretrial Br. 29.

82. Pretrial Br. 29.

83. Pretrial Br. 205.

84. *Ibid.*, 207.

85. Frost & Sullivan Computer Software & Packaged Services Market, Dec. 1977, p. 169.

Software marketing is also maintained by other hardware manufacturers through the use of bundling. A small software company has sued Sperry Rand, charging it with refusing to place a price on its bundled software and thereby making it impossible for the small company to sell its software in competition.[86] Honeywell in bundling its software and hardware asserts that the software is "free." But when the hardware buyer attempts to transfer that software along with the hardware to another company, Honeywell claims that it owns the software and will control who can have possession.[87]

B. The Resulting Poor-Quality Software.

The CBEMA brief also boasts of "The tremendous growth of the computer software industry and the vast number of breakthroughs in the field of programming, which have occurred in the last three decades" (CBEMA Br. 17-19). How different the facts really are in the computer software field is made clear by the testimony of Dr. Ruth M. Davis, former Director of the Center for Computer Services and Technology of the National Bureau of Standards [88] characterizing the present condition of software as "poor-quality software," a condition which she described as the "crucial computer problem" confronting the nation today. Speaking on the "seriousness of the problems besetting both the consumers of computers products and services and the suppliers of these products and services" Dr. Davis stated:

> The crippling problems are lining up in the software rather than in the hardware field. In particular,

86. *Computerworld*, November 20, 1974, p. 45.

87. *Computerworld*, August 6, 1975, p. 1.

88. Testimony given before the Government Activity Subcommittee, House Committee on Government Operations, May 16, 1972.

> they fall under what we call the application-independent problem category . . . these are the software problems which are common to many applications.

This software that is universally used and that strongly affects machine efficiency and that a hardware manufacturer may not always have an incentive to improve are where the "crippling problems" have been. That the condition described by Dr. Davis in 1972 still holds true is seen by her conclusion in 1977:

> "The first 25-year history of software has not been marked with the successes, the advances, or the decreasing costs associated with computer equipment or hardware . . . software is the most unsafe, the least understood, and the most expensive component of total computer system costs.[89] Software development costs are almost 10 per cent of total system costs. . . . The really useful and exciting advances in computing will probably only proceed at the same pace as advances in software engineering. And this will be distressingly so." [90]

A month later Dr. Davis demonstrated her recognition of the innovative nature of software development. At an ADAPSO Management Conference, she stated:

89. Technological developments, especially in the past 15 years, have dramatically reduced the cost of computer circuitry and increased hardware capability. A microprocessor costing about $20 today has the computing power of a large computer that cost $1 million in the 1950's. However, the cost of software has shown the opposite trend; it has risen continuously. The estimated cost of software development, testing and maintenance for the Federal government alone is $4 billion per year while the capital value of government-owned software is about $25 billion.

90. SCIENCE, American Association for the Advancement of Science, Vol. 195, No. 4283, 18 March 1977.

"Software gives us the capability of using computers creatively, flexibly and economically. It is a product that can be designed to benefit users innovatively and uniquely." [91]

Even IBM acknowledges that "the biggest problem facing programming today is the extreme difficulty and cost encountered in creating and maintaining large programming systems." [92] At the National Computer Conference held in 1975, it was admitted by a leading hardware manufacturer that the "software dinosaur" has not "kept pace with the technical needs of the industry." [93] As a result of the software users being at the mercy of the hardware manufacturers, the users were forced to make do with the "free," poor and technically inferior software.[94]

The reason for the undeveloped state of software lies historically in the bundling by CBEMA manufacturers

91. "Customers' Search for Software Substitutes" Keystone Address at ADAPSO's 46th Management Conference "Responding to Change" April 18, 1977.

92. The IBM report from their Poughkeepsie Laboratory—"Composite Design: The Design of Modular Programs" Jan., 1973, p. 3.

93. " 'Software Dinosaur' Slows Growth," Univac Vice President Neil Gorchow, Sperry Rand, *Software Digest*, Vol. 7, No. 20, May 22, 1975, p. 2.

94. "155, 165 Owners Angry With IBM," *Datamation*, August, 1974, p. 76. The "free" software of the hardware manufacturers has been described by a professional for 24 years in electronic data processing (EDP): ". . . Computer systems are in many cases defective. They are primarily defective in the computer programs that come with the hardware—the body of programs referred to as the 'Operating System' . . . a sort of executive in managing the work of the many parts of the computer system, . . . [the] housekeeping chores . . . [A] very big set of programs [that] is often inefficient, overcomplicated and unreliable . . . [t]he outrageous fact remains that the Operating System is often an imperfect and incomplete product which typically contains errors, omissions and inconsistencies. ("EDP—A 20 Year Ripoff," INFOSYSTEMS, 11/74, pp. 26, 27, 29.)

of hardware. History shows that it has been small technological companies and independent inventors which have been responsible for much of the innovation of this age. A recent government report has found that "studies of the sources of invention and innovation have shown that independent inventors and small technologically based companies are responsible for a remarkably high percentage of the important inventions and innovation of this century." [95]

C. The Small Company Potential for Competition.

The small, mostly "young and struggling," companies who "surely possess excellent technical innovative capabilities" and "the principal technical capability . . . to work . . . on our critical computer problems" [96] have been wrongfully denied the incentive of the patent system, which has been assured to the hardware manufacturers for products they sell. The patent incentive is both a form of recognition and a major aid in obtaining the necessary financing without which these companies face the prospect of shriveling and dying. It is no accident that the practices of CBEMA members of marketing by bundling have permitted only a small number of companies in the software products field to achieve a relative degree of success. As the Government states in its pretrial brief "operating systems continue to be part of [hardware manufacturers'] bundled price structure. These continue to be bundled for the marketing leverage they afford [hardware manufacturers'] computer equipment and forestalling the growth of independent software companies." [97]

95. National Bureau of Standards Experimental Technology Inventives Program, Preliminary Program Description, Fiscal Year 1973 Budget Submission, p. 9.

96. Davis, Committee Testimony, p. 15, *supra*.

97. Pretrial Br. 226.

D. The Importance of Patent Incentives.

The need of the incentive of patents for software is as least as great as that of the incentive available for hardware, because: "Today, providing computer software involves greater . . . risk than providing computer . . . hardware. . . ." [98]

To a financial giant, the economic value of a patent may not loom large; to the small software products companies upon which the future of the development of quality software depends, the value of the patent in financing a small company may spell the difference between life and death. To banks and financial institutions the existence of a patent or even the potentiality of obtaining one may well be a decisive factor in determining whether a loan should be granted. To prospective investors a patent or the possibility of obtaining one may be the principal element in the decision whether to invest.

Making clear that patents may be available for inventions in software would unleash important innovative talent. It would have the direct opposite effect forecast by the CBEMA hardware manufacturers; it would enable competition with those companies and provide the needed incentive to stimulate innovation.

Patent protection would encourage the free dissemination of program innovations which is so necessary to stimulate technological progress. Disclosure is the statutory price paid by the inventor for patent protection. Without such protection inventors principally have to rely on the law of trade secrets in order to secure the economic value of their inventions, thereby excluding from the world-at-large disclosure of the invention.

98. COMPUTER SOFTWARE MANAGEMENT. A primer for project management and quality control—Dennis W. Fife. U. S. Dept. of Commerce; NBS Special Publication 500-11, July 1977, p. 1.

When the Patent Office for its own convenience [99] urges that inventors rely on trade secrets in lieu of patents, it is ignoring the mandate of the Patent Laws, and when it urges copyrights as another potential alternative, it is denying the inventiveness of an entire technology, thus abdicating its responsibility under the Patent Laws to grant patents for machines and machine processes, that "advance the Progress of the useful Arts."

Such computer programs as are freely disclosed in the trade literature of the computer industry have little economic value or technological worth. All that this kind of "free exchange" has produced for the computer industry has been "crippling problems" and "poor quality software." The real strides forward in computer development occurred in hardware, an area fully protected by the patent laws. In a competitive economy that which is given away for nothing tends to have the same value. Non-profit companies are non-existent in the computer industry. CBEMA favors its version of "free interchange of programs" because bundling has been one of the major techniques of the hardware manufacturer in retaining for itself the software market for its own hardware. It is profits and not "free interchange of programs" or "innovative software" that CBEMA seeks.

99. In 1966 the Patent Office issued Guidelines which based upon the facts of the technology, provided for the patentability of computer program inventions "Guidelines to Examination of Programs," 829 *Official Gazette*, 441, 442, July 6, 1966. In 1968, the Patent Office issued a set of Revised Guidelines, "Guidelines to Examination of Programs," 855 *Official Gazette*, 829, 830, October 22, 1968, based upon "policy" (*i.e.*, not legal reasons) limiting severely the grant of patents in this area on grounds of administrative convenience, Brenner, *Law of Software, supra*, B-12.

V. EQUAL PROTECTION REQUIRES THAT MANUFACTURERS OF SOFTWARE PRODUCTS BE GIVEN THE SAME PATENT PROTECTION AFFORDED THE HARDWARE MANUFACTURERS.

Petitioner's classification of a softwired special-purpose computer such as respondent's invention as nonpatentable subject matter lacks any rational basis in fact; hence, it violates respondent's equal protection rights under the Fifth Amendment.[100] Respondent may not be denied patent protection for his machine invention solely because it is software-embodied absent some rational basis for the discrimination. Arbitrary economic favoritism is at the heart of the denial of patents to inventors of software-embodied special-purpose computers while granting patents to inventors of hardware-embodied special-purpose computers.[101]

The instant case is strong, because the patent statute is by its terms generally inclusive.[102] This is not a case in

100. While the Fifth Amendment does not contain an explicit Equal Protection clause, Equal Protection standards applicable to the Federal Government have been incorporated into the Due Process clause of the Fifth Amendment. See *Richardson v. Belcher*, 404 U. S. 78, 81 (1971); *Bolling v. Sharpe*, 347 U. S. 497, 499 (1954).

101. Three different comparisons describe the discrimination against software. First, the Commissioner would grant a patent on the hardware embodiment of a particular special-purpose computer but not on the equivalent software embodiment of the very same machine. Second, the Commissioner would grant patents for the innovative machine products of their competitors, the small software companies. Third, while machines as a class are patentable, 35 U. S. C. § 101, the Commissioner has singled out software-embodied machines and control processes and no others are unpatentable. These discriminations are all predicated on a myth of the non-machine—the bald assertion that a machine constructed with software is an "idea" in direct conflict with the rulings of the Court; see *Burr v. Duryee*, 68 U. S. 531.

102. "[W]hoever invents . . . *any* new and useful . . . machine . . . may obtain a patent therefor. . . ." 35 U. S. C. § 101 (emphasis added). The only exception from patentability of a

which great deference need be given a studied intentional congressional act of exclusion since there was none. To the contrary, the patent policy urged by the Government here violates the Small Business Act[103] in that it discriminates against the small businesses which form the heart and core of the software products industry. Such an inconsistent action by the Government should certainly be predicated upon a sound legal basis rather than a Luddite[104] policy of opposing technological advances. No judicial sanction should be given a strained administrative interpretation that runs counter to the clear import of the statute. While an action of the Commissioner that is consistent with an explicit congressional directive is entitled to deference, when he acts on his own discretion and without congressional support, "a very different question is presented." *Boyden v. Comm'r. of Patents*, 441 F. 2d 1041, 1044 (D. C. Cir. 1971).

102. (Cont'd.)
specific machine technology is that of atomic weapons; see 42 U. S. C. § 2181; but Congress otherwise provided for incentive compensation 42 U. S. C. § 2187(b), (3). There Congress made explicit the exception, reinforcing its mandate that all other machine technologies were intended to be patentable subject matter.

103. 15 U. S. C. § 631, provides: "It is the declared policy of the Congress that the Government should aid, counsel, assist, and protect, insofar as is possible, the interests of small-business concerns in order to preserve free competitive enterprise . . . and to maintain and strengthen the over-all economy of the Nation."

104. "We are probably now at about the comparable point in time in the Computer Revolution to that in the Industrial Revolution when the Luddites were vainly smashing equipment." H. H. Goldstine, *supra*, p. 347. For the last decade and more, there have been hysterical arguments raised against the patenting of software-embodied inventions. First Amendment questions raising the spectre of complete computer domination, panicked fears of the undermining of the patent system, concocted theories of machines being "ideas" and of administrative convenience, have been among the many short-lived stratagems put forth to deny recognition to the uncontrovertible advances of software technology. Must we be so fearful of humankind's ability to utilize technology beneficially that it is necessary to stand against further technological advancement by denying its existence and discouraging its forward thrust.

The Commissioner of Patents offers no justification related to the purposes of thet Patent Act for his classification of software inventions as unpatentable. He does not deny that all machines and machine control processes are patentable subject matter. Instead, the petitioner defends the discrimination with the bald mischaracterization of the programmed general-purpose computer as an "idea" and not a new machine, nor a "specifically devised apparatus" (Br. Pet. 14). This strained unrealistic attempt to rule out an entire machine technology from the patent laws cannot withstand scrutiny. While the petitioner's brief disregards the machine nature of programmed general-purpose computers, he has not presented one bit of supporting evidence to any tribunal in this case, or in any of the predecessor "computer programming" cases.

When the Patent Commissioner singles out for exclusion the economically struggling segment of the data processing industry, and has favored the "economic" interests of the "collective forces of major hardware (i.e., computer) manufacturers and their representative associations," [105] judicial protection against unjustified discrimination is especially necessary. That this is so, is made painfully clear by the bundling condition which exists in the computer industry and which the Government admits has proved so harmful to the software products industry and to our national interests (see Nature of the Software Products Industry, *supra* p. 43).

The form of construction of a special-purpose computer—with hardware or software—bears no relationship to the patent law or any other legitimate government interest. The "useful Arts" are promoted equally by hardware and software patents. In fact, the exclusion of software patents has had a severely detrimental effect on the functioning of the patent system. Since hardware and

105. Judge Rich in the *Johnston* case 502 F. 2d 765, 774.

software can be alternative embodiments of the same machine structure, it is necessary to conduct a search of the prior software art to determine whether a particular hardware invention is novel.[106] The lack of software patents means that there have been inadequate search facilities for hardware.[107] The CBEMA hardware manufacturers seek patents for their hardware but they do not advocate patents for software-embodied machines that would become part of the search resources.

Different treatment based on irrelevant distinctions is unconstitutional. In this case, since invention can take place equally in hardware and software (the choice of construction often being a matter merely of "trade-offs") and since the two embodiments are technologically equivalent, both must be treated identically for purposes of the patent law.

106. The 1966 Report of the President's Commission on the Patent System (reprinted in S. Doc. No. 5, 90th Cong., 1st Sess.; p. 21, now long outdated, reflected the inaccurate and one-sided view of the hardware manufacturers that the software state of the art is irrelevant to hardware. It is to be regretted that the President's Commission permitted itself to be used as a vehicle for the dissemination of this technologically unsound point of view. Much of the innovation in hardware was and still is first achieved in software. Software prior art is as essential as the hardware prior art to adequate examination of the novelty of a hardware-implemented invention. Similarly, both are needed for examination of an equivalent software-implemented invention.

107. Delinquent in its duties for many years, the Patent Office finally in 1972 "initiated a pilot program, with the cooperation of the Bureau of Standards, to develop techniques for classification and searching of the computer programming art." (*See* Pet. Br. *Benson*, p. 31). Since no longer is the argument being raised by the Patent Office about inadequate search facilities, one must conclude that the problem has been solved. The tragedy is that so-called "search facilities" were permitted to become a problem at all. Only the hardware manufacturers benefited during the interim period because it meant that the Examiners were being deprived of the proper research facilities to evaluate the so-called "hardware" inventions, thereby seriously undermining the entire credibility of the patent system. Fortunately, the Patent Office has apparently, albeit belatedly, taken corrective action.

The Government's suggestion of alternative modes of protection for software innovations, such as trade secrets or copyrights,[108] are inadequate and inappropriate to the task, and destructive of the constitutional purpose of the patent system as an incentive to machine innovation.[109] There is no evidence of any intent of Congress to relegate software inventors to reliance on other means of protection.

Copyright protects particular expression, not machine structure. There is no requirement of innovation, only originality; therefore, a copyright would not protect the machine configuration that is at the heart of respondent's invention. The fact that a "program" can be represented as a list of computer operations is no more significant than the fact that a radio circuit can be represented in a schematic diagram. An innovative radio circuit would not be protected by a copyright on a drawing and the special-purpose computer that is respondent's invention would not be protected by a copyright on a program listing.

Trade secrecy is not only inadequate, but forced reliance on its violates the patent policy of disclosure which is needed to advance the "useful Arts". To the inventor, it is burdensome and often impossible to assure the complete secrecy that is required to retain the value of the innovation. Beyond this, reliance on this mode of protection has the counterproductive effect, opposed by

108. Here, too, with respect to copyrights, the 1966 President's Commission allowed itself to be used as an instrument to becloud the issue. The recommendations of the 1966 President's Commission seeking to deny patentability to software-implemented inventions were soundly rejected by Congress; see H. R. 5924, proposed, 35 U. S. C. § 106, 90th Cong., 1st Sess. (1967).

109. The "poor-quality software" that presently exists and the need for the incentive of the patent system is described above at pp. 40-45.

public policy, of deterring disclosure of software innovations, hence, inhibiting the progress of the "useful Arts." Neither trade secrecy nor copyright is a substitute for equal treatment under the patent law.

In sum, the petitioner's arbitrary refusal[110] to grant patents for software inventions, in addition to violating the clear command of the statute, is a deprivation of the inventor's equal protection rights. The petitioner offers no rational reasons for excluding from patentable subject matter the software machine technology. Petitioner's only answer to the equal protection dilemma is the bald assertion that the invention does not include "a specifically devised apparatus" (Br. Pet. 14). Without any evidentiary support, contrary to the views of both the Board and the C. C. P. A., and not jurisdictionally before this Court, that assertion cannot justify nor sustain an unconstitutional Patent Office policy of wholly irrational and unequal treatment.

Software is a new technology that is today viewed as the most critical and important element of a computer system. As in any new technology, for it to grow and compete, it needs all the protection due it under the law. ADAPSO asks the Court to treat the software technology equal to the hardware technology—no better—and no worse.

110. In 1968, the then-Commissioner of Patents stated: "The present Office policy of refusing patents in this [computer program] field has acted as a strong deterrent to the filing of applications." (Commissioner Brenner, *supra*, p. B-12) By its action, the Patent Office acted legislatively in opposition to the patent statutes and to the Small Business Act. While the most direct sufferers of this Patent Office policy are the software products companies engaged in innovative machine building, it is a fact that the loss is suffered by the whole nation which is deprived of the knowledge that would be disclosed to it by the award of a patent.

CONCLUSION.

For the reasons stated above, it is respectfully urged that the writ of certiorari be dismissed. In the alternative, if this Court reaches the merits of the case, the decision of the Court of Customs and Patent Appeals on patentable subject matter should be affirmed.

Respectfully submitted,

MORTON C. JACOBS,
DAVID COHEN,
Attorneys for Amicus ADAPSO.